School Of Engineering

Electronics and Communication Engineering

TITLE OF THE PROJECT :

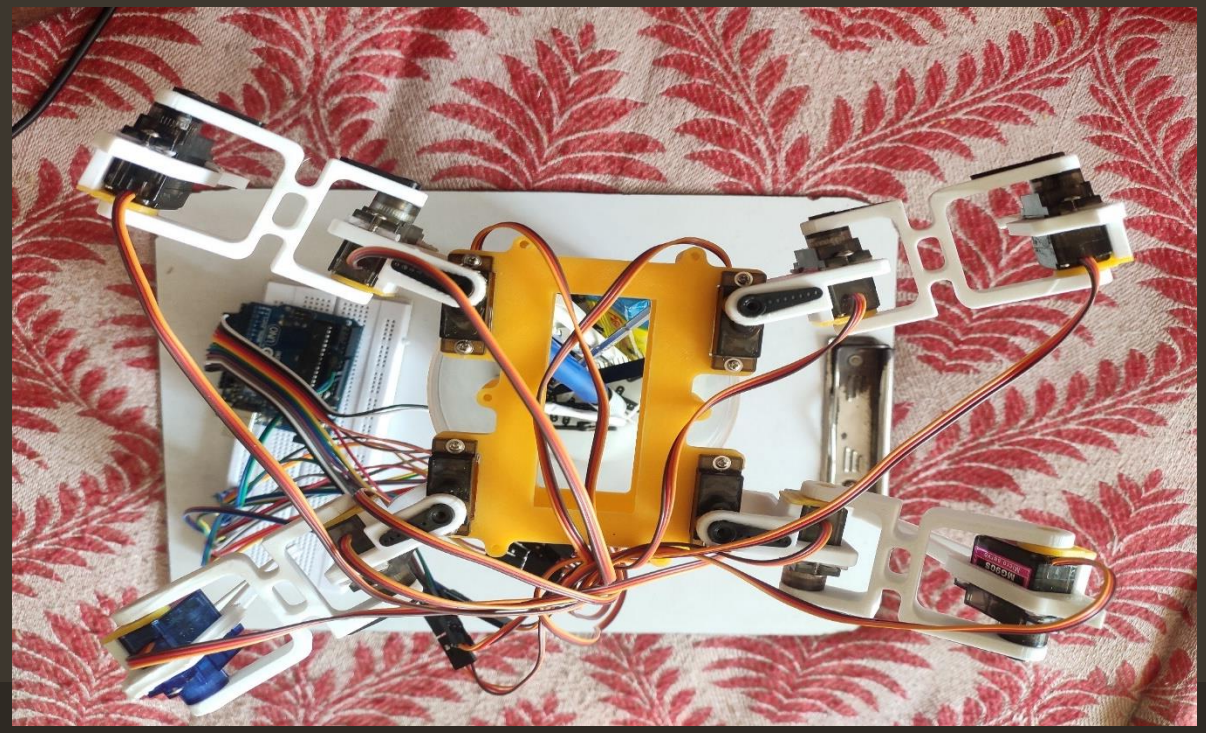
Quadra-Bot using Raspberry Pi

Presented By : Pilli Venkata Praneeth
Roll No : 2019BEC22
Semester : 8th

Submitted to: Dr. Layak Ali
Assistant Professor.

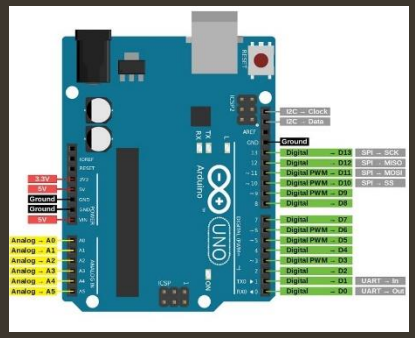
Introduction

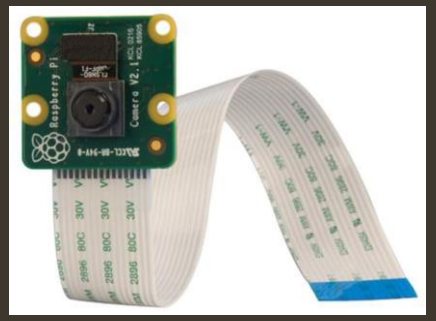
- Quadra-Bot is a cutting-edge robot that has been designed to perform various tasks with precision and accuracy. It is equipped with advanced sensors and intelligent algorithms that enable it to navigate through complex environments and carry out its functions efficiently.
- The development of Quadra-Bot has been a significant breakthrough in the field of robotics. Its unique design and capabilities have made it an ideal solution for many industries, including manufacturing, healthcare, and logistics.

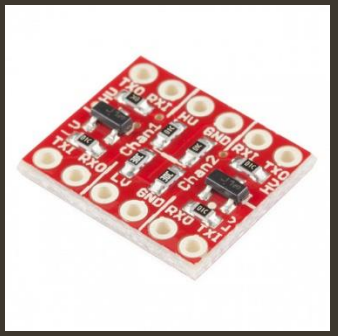


Components used :

- Raspberry Pi 3B
- Arduino Uno
- Mg90S servo motors-12
- Pi-Camera module
- logic level shifter
- 3.7v rechargeable li-ion battery
- Buck convertor LM2596







Objective

1. Mobility: Four-legged robots are often designed to move over uneven terrain and in environments that are challenging for wheeled or tracked robots.

2. Stability: Four-legged robots have a low center of gravity and a wide base, which can provide them with greater stability than other types of robots.

3. Maneuverability: Four-legged robots can navigate through narrow spaces and complex environments with greater ease than larger wheeled or tracked robots.

4. Payload Capacity: Some four-legged robots are designed to carry payloads, such as sensors or tools, which can be used for a variety of tasks, such as inspection, search and rescue, or transportation.

5. Versatility: Four-legged robots can be designed for a variety of applications, including military, industrial, and research, making them a versatile and adaptable technology.

Manufacturing Applications of Quadra-Bot

- In the manufacturing industry, Quadra-Bot has proven to be a valuable asset. Its ability to perform multiple tasks simultaneously has increased productivity and reduced production time. It can assemble components, weld, paint, and inspect products with high accuracy and consistency.
- Furthermore, Quadra-Bot's advanced sensors and algorithms allow it to detect defects and errors in the manufacturing process, preventing costly mistakes and ensuring product quality.

Healthcare Applications of Quadra-Bot

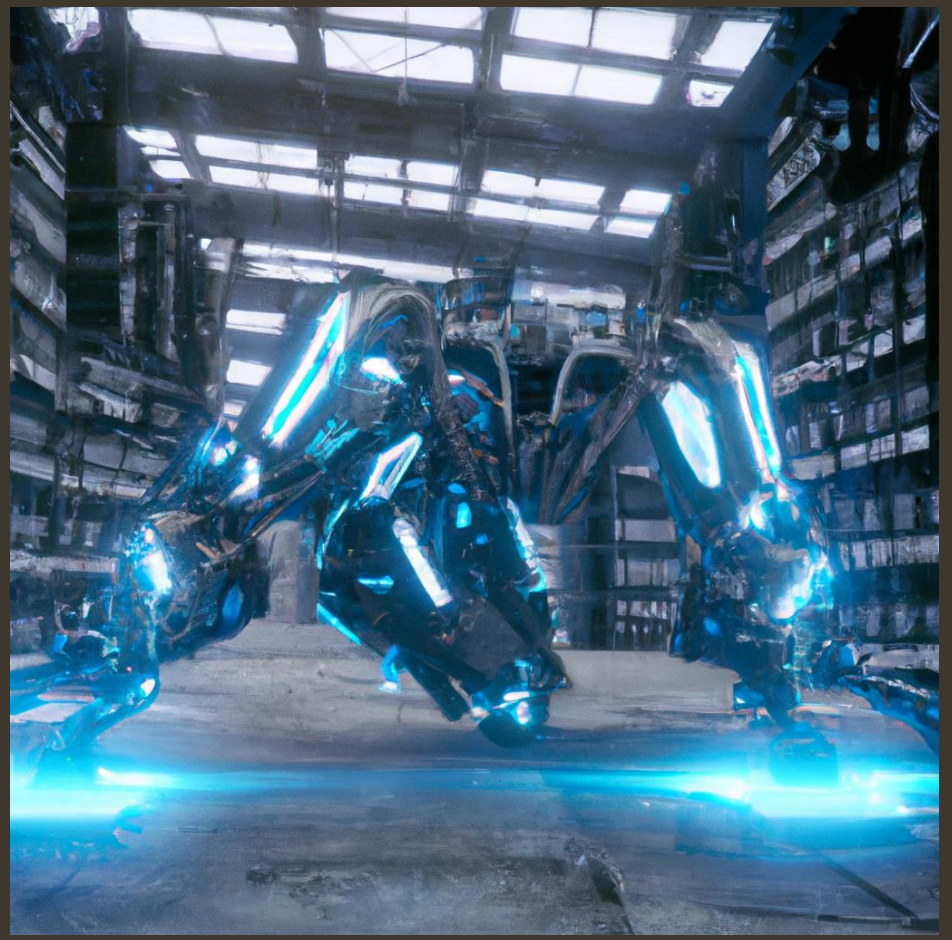
- Quadra-Bot has also found applications in the healthcare industry. Its ability to perform delicate procedures with precision and accuracy has made it an ideal tool for surgeries and medical interventions.
- Additionally, Quadra-Bot's advanced sensors and algorithms allow it to monitor patients' vital signs and provide real-time feedback to healthcare professionals, improving patient outcomes and reducing the risk of complications.

Logistics Applications of Quadra-Bot:

- Quadra-Bot's versatility and precision also make it an ideal solution for logistics and warehouse management. It can efficiently move and stack boxes, pick and pack products, and even load and unload trucks.
- Furthermore, Quadra-Bot's advanced sensors and algorithms allow it to navigate through complex environments and avoid obstacles, reducing the risk of accidents and improving safety in the workplace.

Future Developments of Quadra-Bot

- As technology advances, so will Quadra-Bot. Future developments may include enhanced sensory capabilities, improved algorithms, and increased mobility.
- Additionally, Quadra-Bot may find applications in new industries, such as space exploration and underwater research, where its unique design and capabilities could prove invaluable.



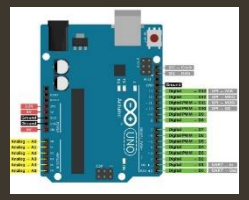
Block Diagram

Servo-1

Servo-2

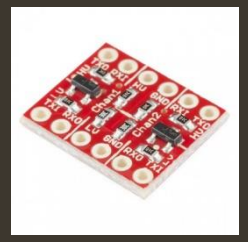
Servo-1

Servo-12

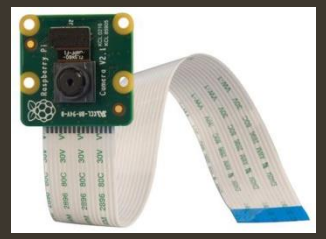


Arduino-Uno

Logic level shifter



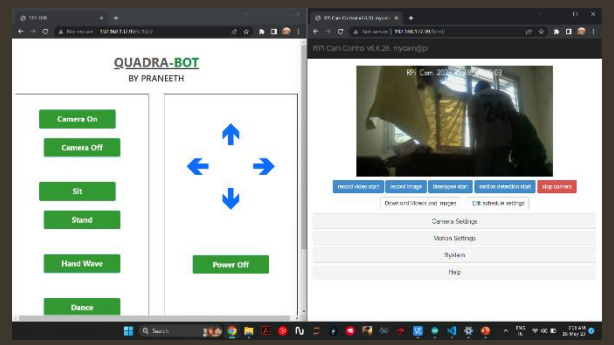
Raspberry Pi-3



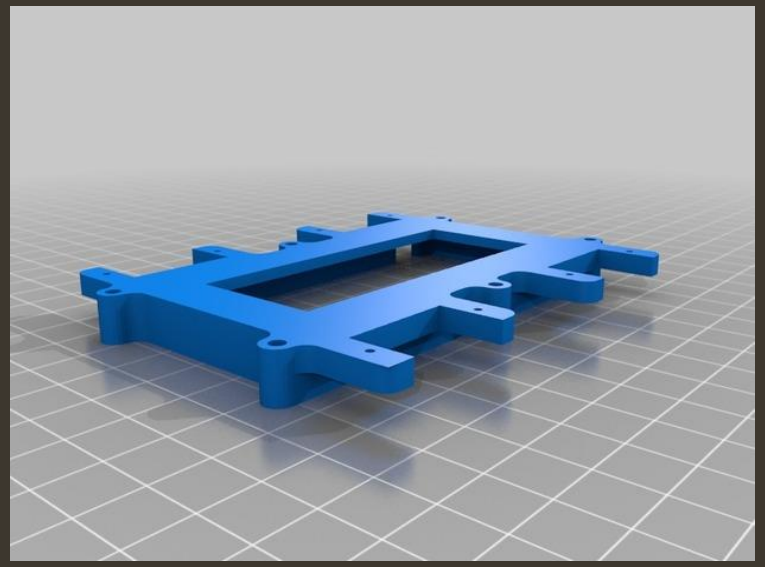
Camera module

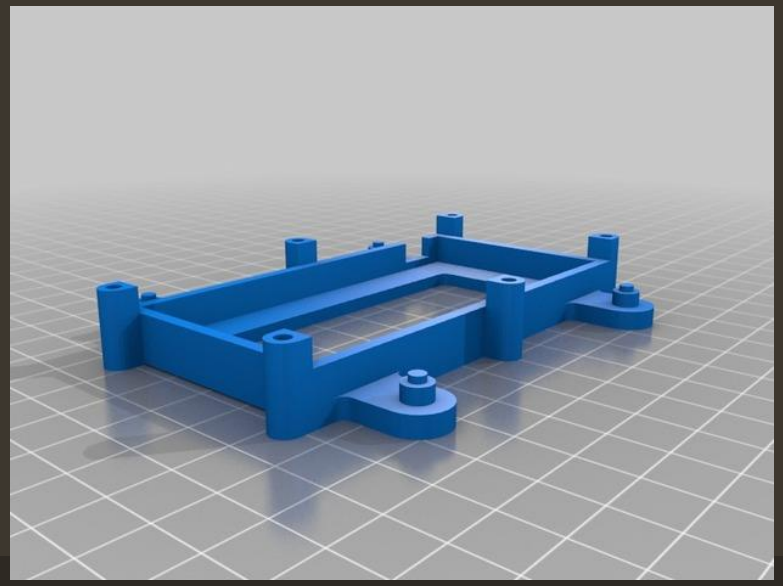
WiFi-hotspot connection

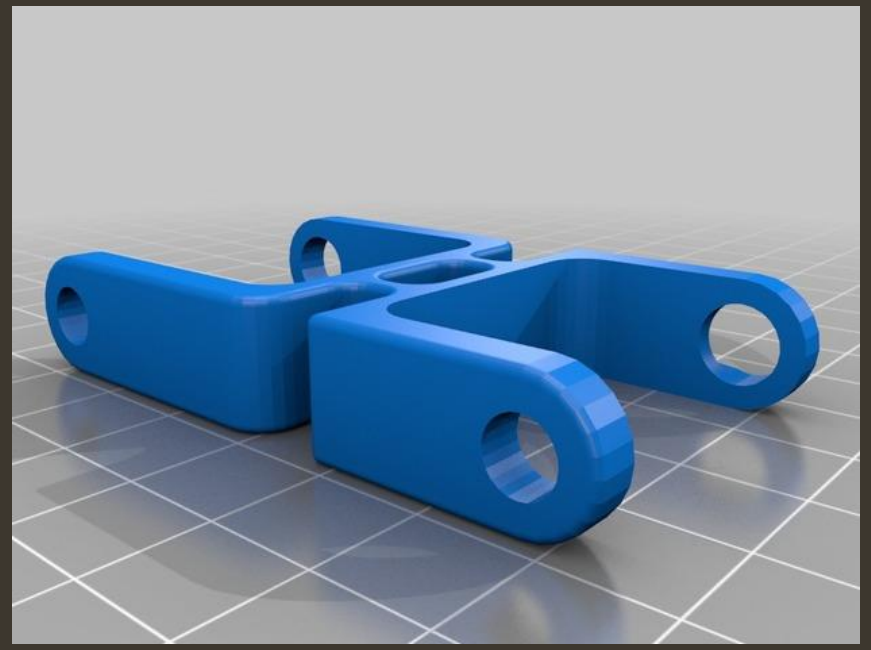
Web-controlling

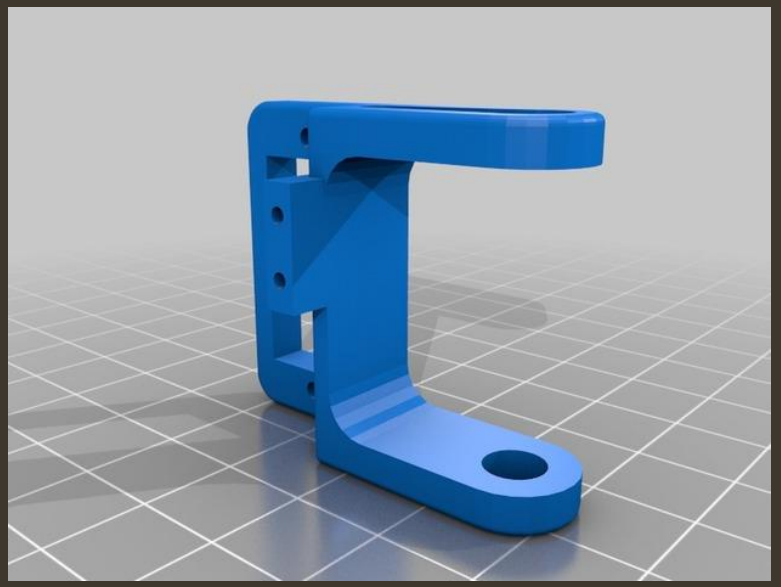


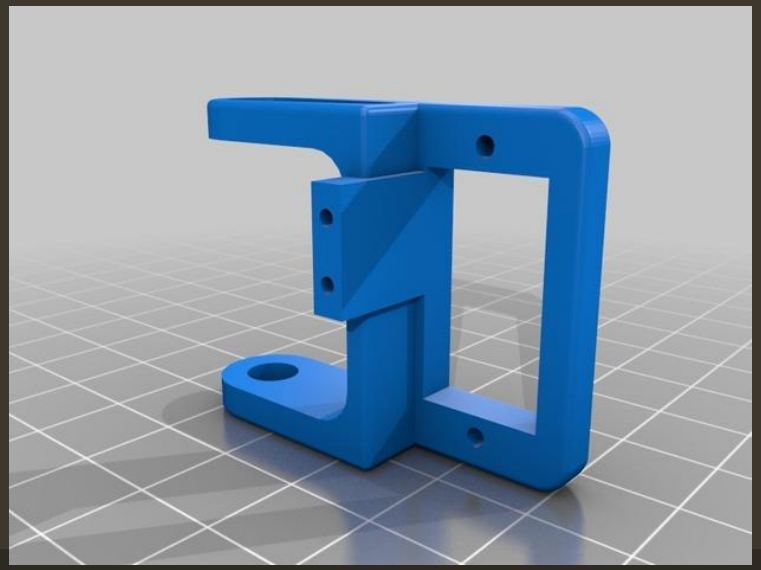
Designing of bot Body



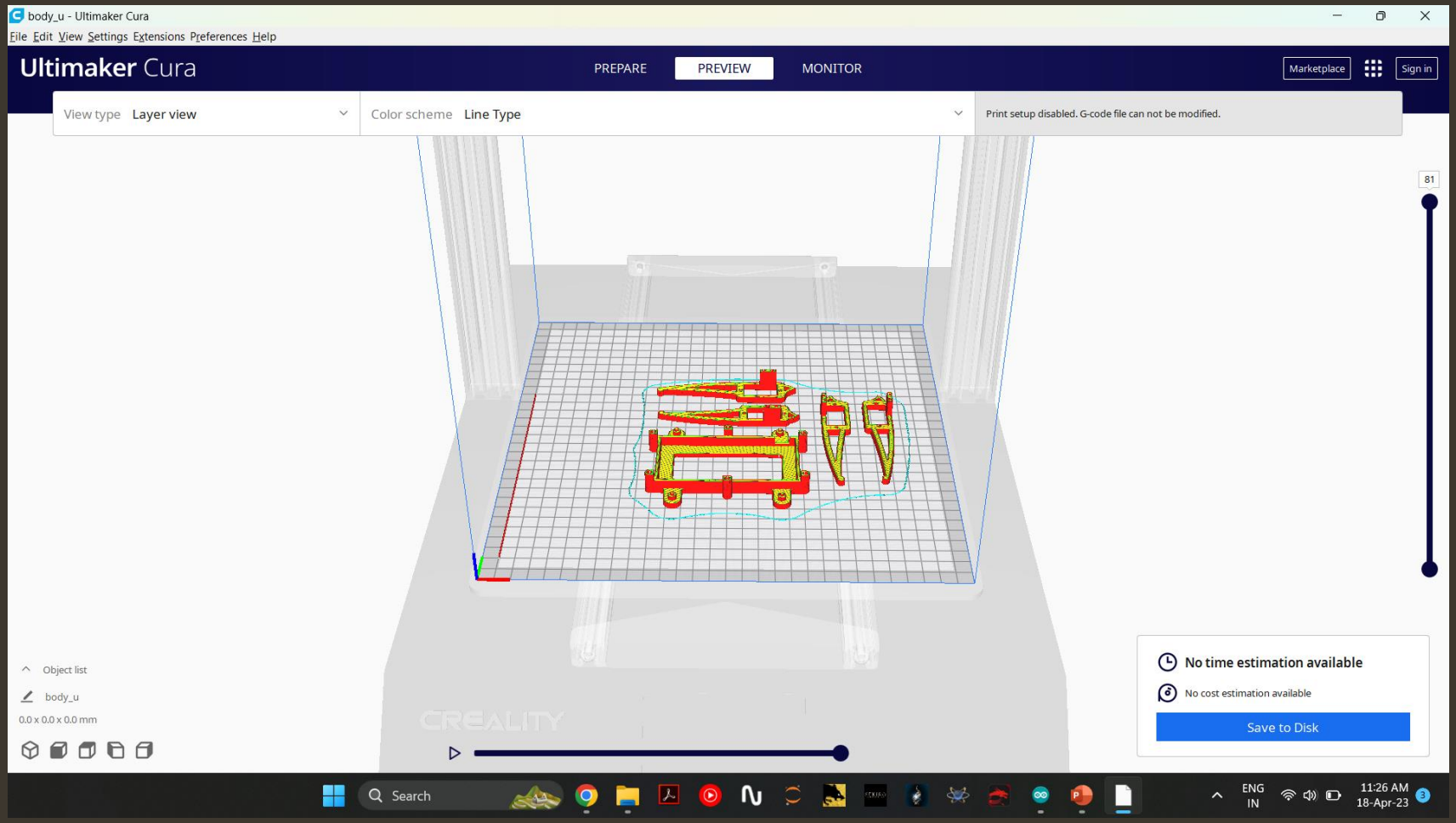




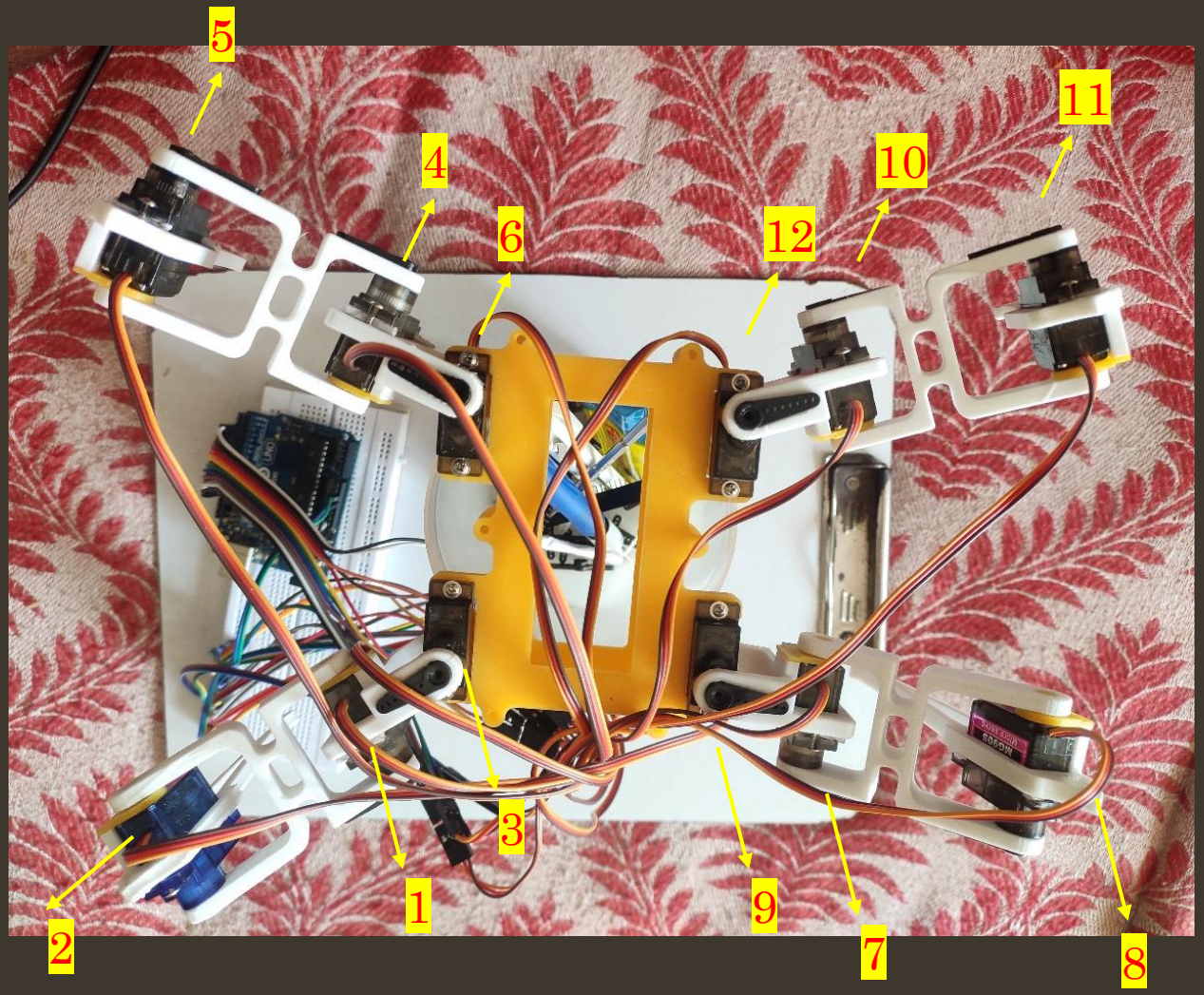




3D printing of designed Body



Implementation



Legs | Arduino 1.8.10

File Edit Sketch Tools Help

Legs

```
#include <Servo.h>

Servo servo[4][3];

//define servos' ports
const int servo_pin[4][3] = { {2, 3, 4}, {5, 6, 7}, {8, 9, 10}, {11, 12, 13} };

void setup()
{
  //initialize all servos
  for (int i = 0; i < 4; i++)
  {
    for (int j = 0; j < 3; j++)
    {
      servo[i][j].attach(servo_pin[i][j]);
      delay(20);
    }
  }
}

void loop(void)
{
  for (int i = 0; i < 4; i++)
  {
    for (int j = 0; j < 3; j++)
    {
      servo[i][j].write(90);
      delay(20);
```

1

Search

Arduino code

Legs | Arduino 1.8.10

File Edit Sketch Tools Help

Legs

```
#include <Servo.h>

Servo servo[4][3];

//define servos' ports
const int servo_pin[4][3] = { {2, 3, 4}, {5, 6, 7}, {8, 9, 10}, {11, 12, 13} };

void setup()
{
  //initialize all servos
  for (int i = 0; i < 4; i++)
  {
    for (int j = 0; j < 3; j++)
    {
      servo[i][j].attach(servo_pin[i][j]);
      delay(20);
    }
  }
}

void loop(void)
{
  for (int i = 0; i < 4; i++)
  {
    for (int j = 0; j < 3; j++)
    {
      servo[i][j].write(90);
      delay(20);
```

1

Search

SPY-DER_Arduino | Arduino 1.8.10

File Edit Sketch Tools Help

SPY-DER_Arduino | FlexiTimer2.cpp | FlexiTimer2.h

```
{
  //start serial for debug
  Serial.begin(9600);
  Serial.println("Robot starts initialization");
  //initialize default parameter
  pinMode(0, OUTPUT);
  set_site(0, x_default - x_offset, y_start + y_step, z_boot);
  set_site(1, x_default - x_offset, y_start + y_step, z_boot);
  set_site(2, x_default + x_offset, y_start, z_boot);
  set_site(3, x_default + x_offset, y_start, z_boot);
  for (int i = 0; i < 4; i++)
  {
    for (int j = 0; j < 3; j++)
    {
      site_now[i][j] = site_expect[i][j];
    }
  }
  //start servo service
  FlexiTimer2::set(20, servo_service);
  FlexiTimer2::start();
  Serial.println("Servo service started");
  //initialize servos
  servo_attach();
  Serial.println("Servos initialized");
  Serial.println("Robot initialization Complete");
  stand();
}


void servo_attach(void)
```

Updates available for some of your boards and libraries

Search

SPY-DER_Arduino | Arduino 1.8.10

File Edit Sketch Tools Help

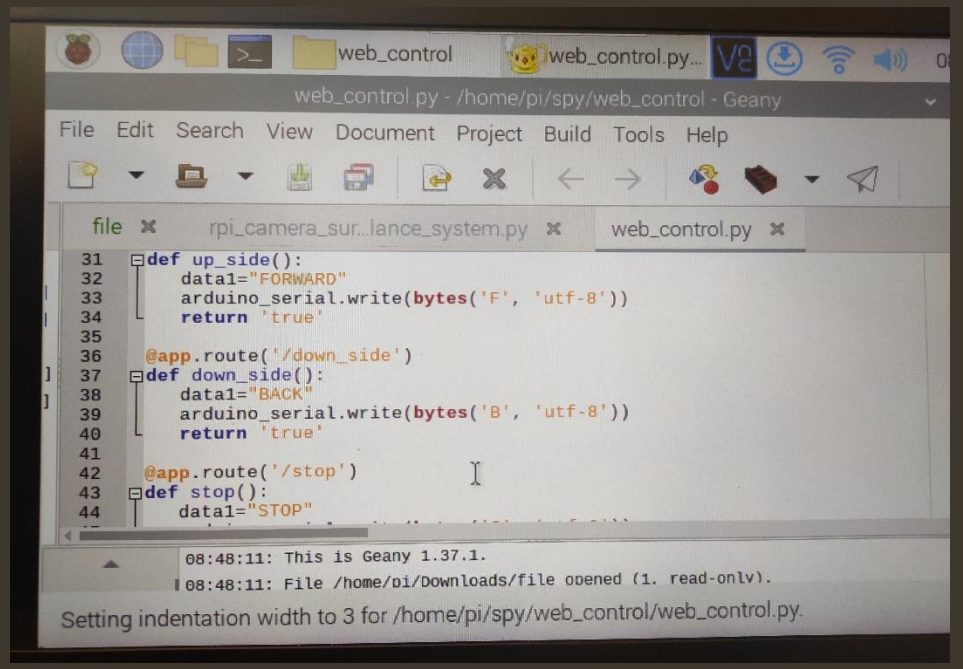
SPY-DER_Arduino | FlexiTimer2.cpp | FlexiTimer2.h

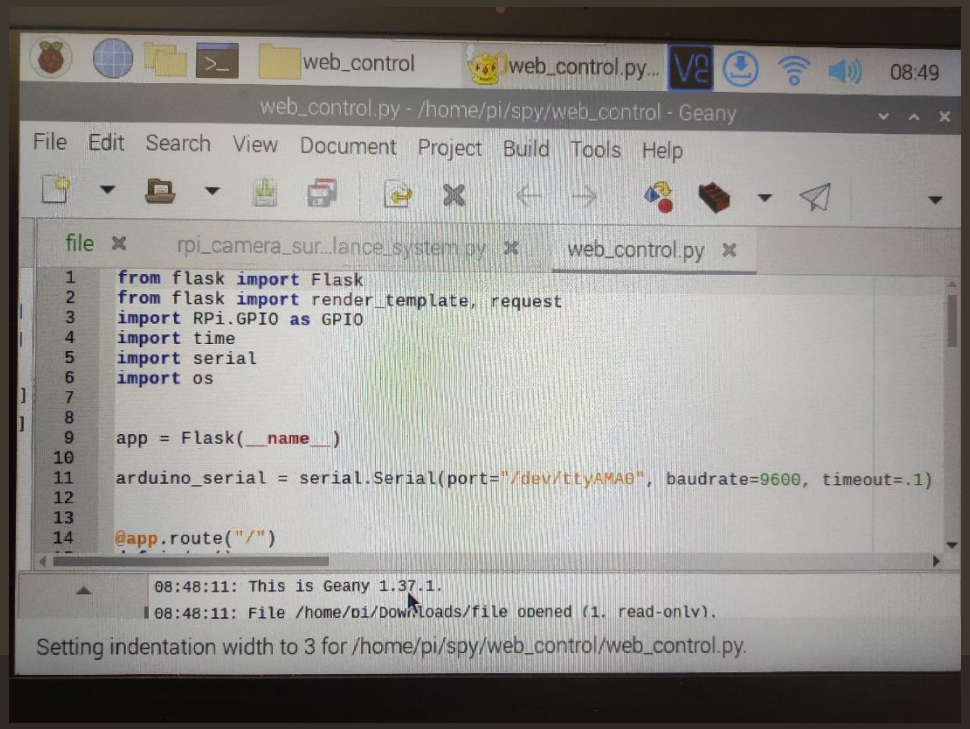
```
if(Serial.available() > 0)
 {
    data = Serial.read();
    Serial.print(data);
    Serial.print("\n");
    if(data == 'f')
      {
       Serial.println("Step forward");
       step_forward(5);
      }
    else if(data == 'b')
       {
        Serial.println("Step back");
        step_back(5);
      }
    else if(data == 'l')
       {
       Serial.println("Turn left");
       turn_left(5);

      }
    else if(data == 'r')
      {
       Serial.println("Turn right");
       turn_right(5);

      }
```

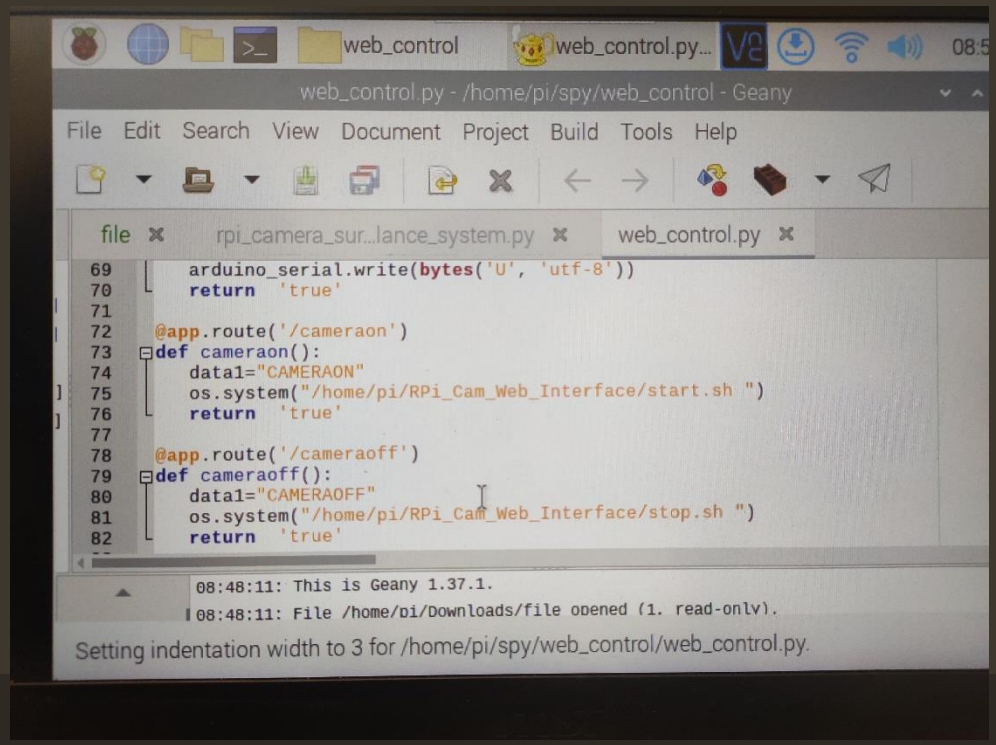
1

Search

Web control using Raspberry- pi 3



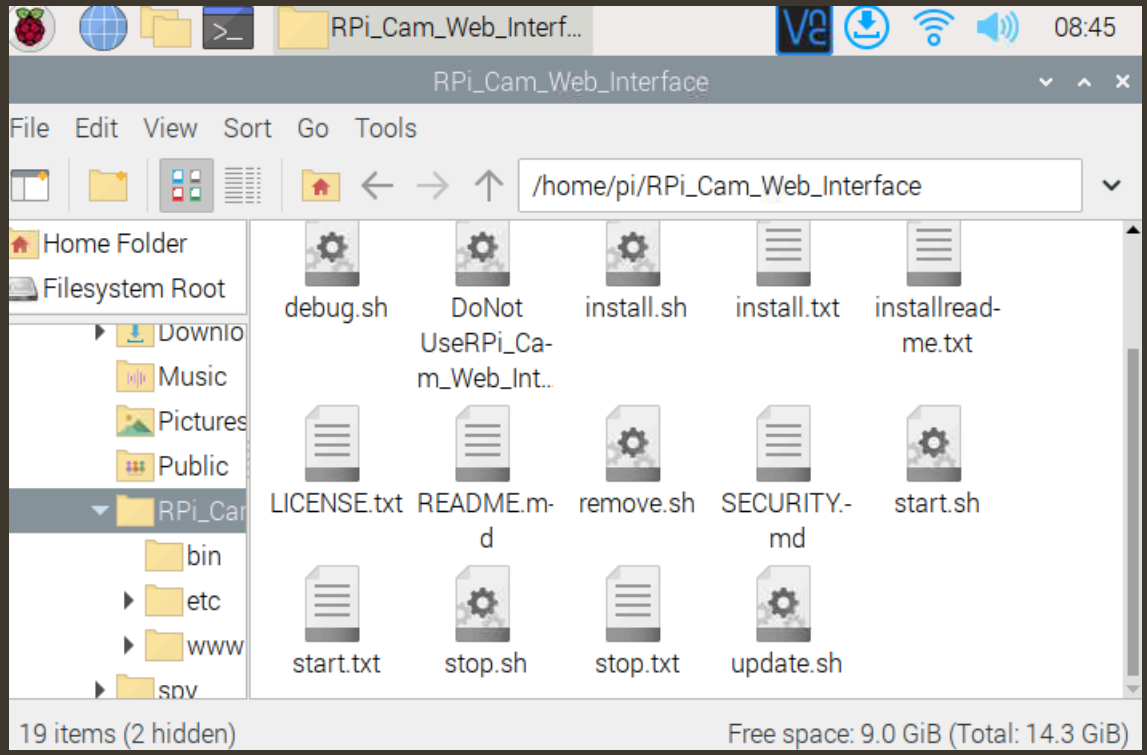




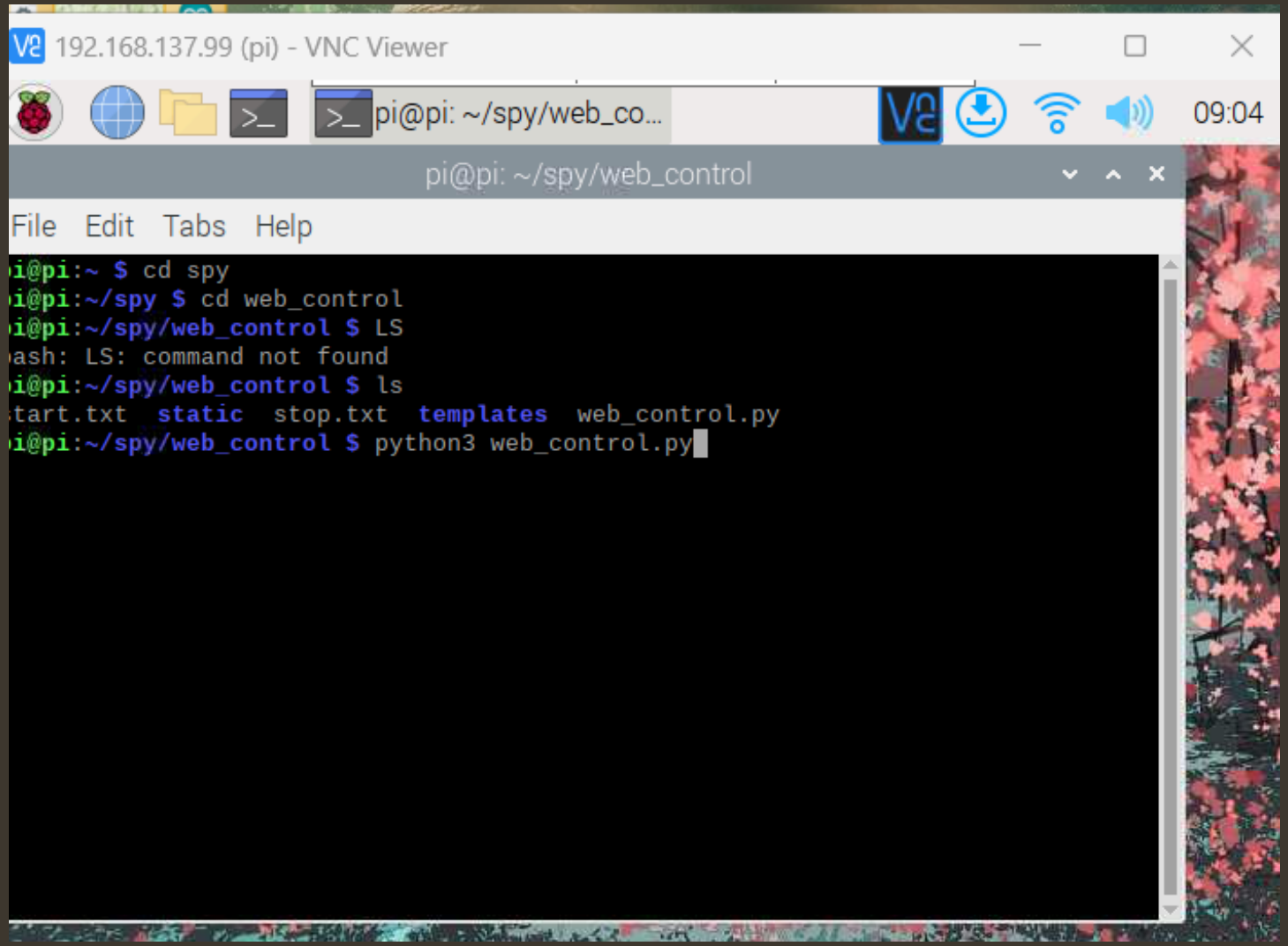
Video surveillance

RPi Cam Web Interface is a web interface for the Raspberry Pi Camera module. It can be used for a wide variety of applications including surveillance, dvr recording and time lapse photography. It is highly configurable and can be extended with the use of macro scripts. It can be opened on any browser (smartphones included) and contains the following features:

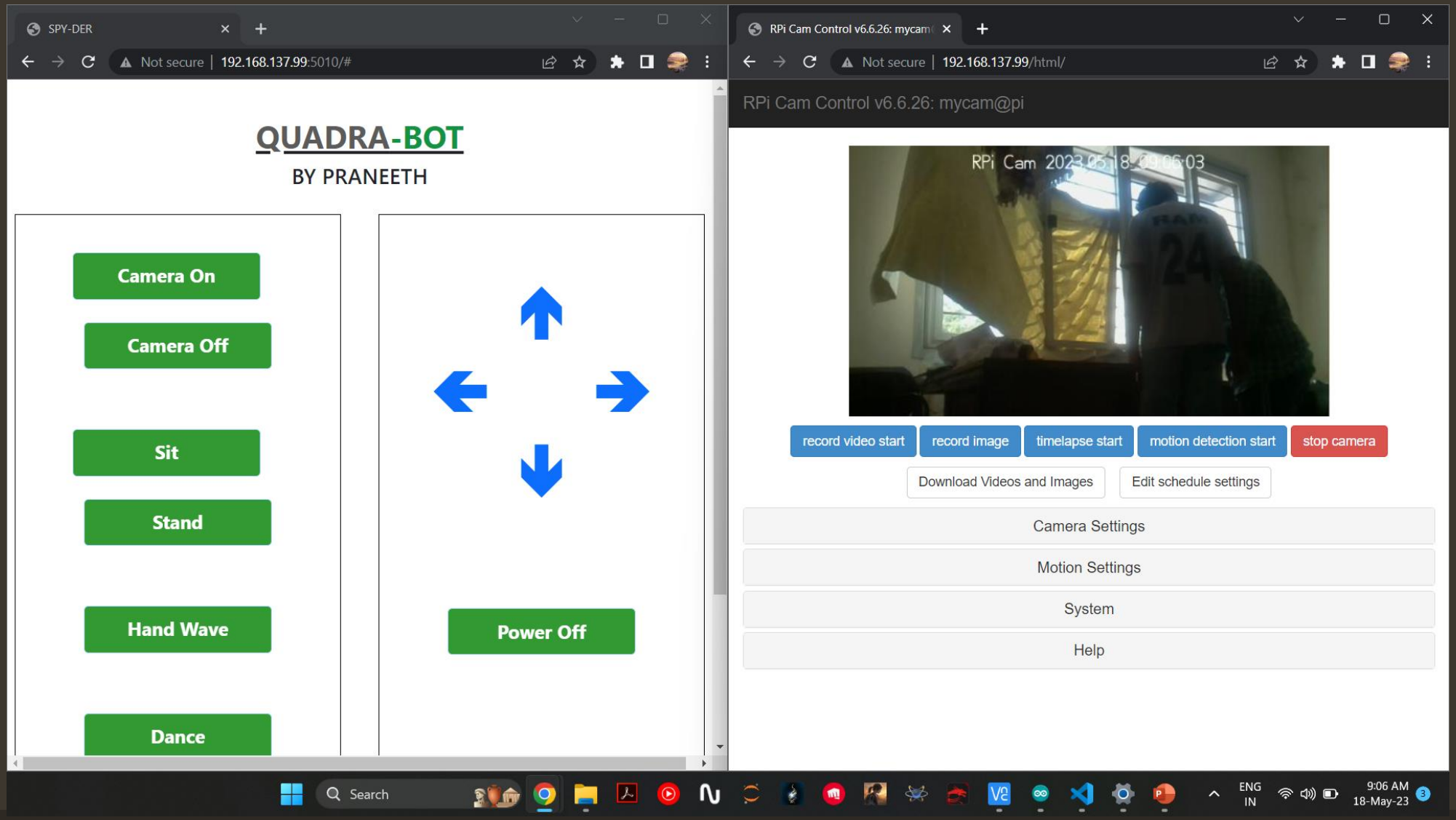
•View, stop and restart a live-preview with low latency and high framerate. Full sensor area available.
•Control camera settings like brightness, contrast, ... live
•Record full-hd videos and save them on the sd-card packed into mp4 container while the live-preview continues
•Do timed or continuous video recording with splitting into fixed length segments
•Take single or multiple (timelapse) full-res pictures and save them on the sd-card (live-preview holds on for a short moment)
•Preview, download and delete the saved videos and pictures, zip-download for multiple files



Raspberry Commands



Interfacing web-control and Pi-camera




Thank
You